Multi-Mode Tensor Space Clustering Based on Low-Tensor-Rank Representation

Yicong He[1], **George K. Atia**[1,2]

[1]Department of Electrical and Computer Engineering, University of Central Florida
[2]Department of Computer Science, University of Central Florida
{yicong.he, george.atia}@ucf.edu

Abstract

Traditional subspace clustering aims to cluster data points lying in a union of vector subspaces. The vectorization of multi-dimensional data to perform clustering disregards much of the structure intrinsic to such data. To capture said structure, in this work we perform clustering in a high-order tensor space rather than a vector space. We develop a novel low-tensor-rank representation (LTRR) for unfolded matrices of tensor data lying in a low-rank tensor space. The representation coefficient matrix of an unfolding matrix is tensorized to a 3-order tensor, and a low-tensor-rank constraint is imposed on the resulting coefficient tensor to exploit the self-expressiveness property. Then, inspired by the multi-view clustering framework, we develop a multi-mode tensor space clustering algorithm (MMTSC) that can deal with tensor space clustering with or without missing entries. The tensor is unfolded along each mode, and the coefficient matrices are obtained for each unfolded matrix. The low-tensor-rank constraint is imposed on a tensor constructed from transformed coefficient tensors for each mode, thereby simultaneously capturing the low rank property of the data within each tensor space and maintaining cluster consistency across different modes. Experimental results demonstrate that the proposed MMTSC algorithm can in many cases outperform existing clustering algorithms.

Introduction

Subspace clustering has received strong interest in recent years due to its wide applicability in diverse domains, including computer vision (Zhang, Jiang, and Davis 2013; Wang et al. 2018a), image processing (Hong et al. 2006; Zhang et al. 2016), and bioinformatics (Zheng et al. 2019; Peignier, Rigotti, and Beslon 2015). The theoretical underpinning of subspace clustering is that high-dimensional data points lie in a union of low-dimensional subspaces (Vidal 2011), which has led to numerous successful clustering algorithms in the last decades (Parsons, Haque, and Liu 2004; Elhamifar and Vidal 2013; Gao et al. 2015). Spectral clustering, in which points are clustered by partitioning a similarity graph derived from the data (Von Luxburg 2007), is the basis for some of the most popular and modern clustering methods, such as sparse spectral clustering (SSC) (Elhamifar and Vidal 2013), low rank representation (LRR) (Liu et al. 2010) and least squares regression (LSR) (Lu et al. 2012).

In traditional subspace clustering, the data is assumed to lie close to linear (vector) subspaces. To cluster multi-dimensional data, a common approach is to first map it to vectors (vectorization), then apply subspace clustering to cluster the points into groups that belong to the same subspace (e.g., see (Vidal 2011) for a survey.) A drawback of this approach, however, is that it does not fully exploit multi-way correlations intrinsic to such data (Piao et al. 2016; Fu et al. 2016). For example, in video sequences, frames could have strong correlations in the temporal domain, and vectorization often falls short of capturing the spatial and temporal correlations among the adjacent pixels.

Recently, (Ashraphijuo and Wang 2020) presented a theoretical study of the problem of clustering tensors drawn from a union of low-rank tensor spaces. This work brought to the fore the utility of employing multi-dimensional data structures when performing clustering, since the number of data points and the sampling rate requirements to guarantee correct clustering based on tensor analysis were shown to be significantly lower than with matrix analysis applied to unfoldings of the tensor data. However, (Ashraphijuo and Wang 2020) did not develop actual algorithms for tensor space clustering, which is the primary focus of our work.

In this paper, we develop a novel low-tensor-rank representation (LTRR) and a multi-mode tensor space clustering (MMTSC) algorithm to cluster data lying in a union of tensor spaces with or without missing entries. By leveraging a tensor circular unfolding scheme (Yu et al. 2018, 2019), the tensor can be unfolded into several matrices along different modes and the corresponding representation coefficient matrices are obtained. In sharp contrast to SSC and LRR which impose a sparse and low rank constraint on the representation coefficient matrices, we impose a low-tensor-rank constraint on the transformed coefficient tensors to capture the self-expressiveness property of the tensor data. Then, a multi-mode clustering algorithm is proposed, which simultaneously captures the low-tensor-rank property for each coefficient tensor and the consistency of clustering across the different modes. The main contributions of this paper are summarized as follows:

- We propose a novel low-tensor-rank representation for unfolding matrices of high-dimensional data lying in a

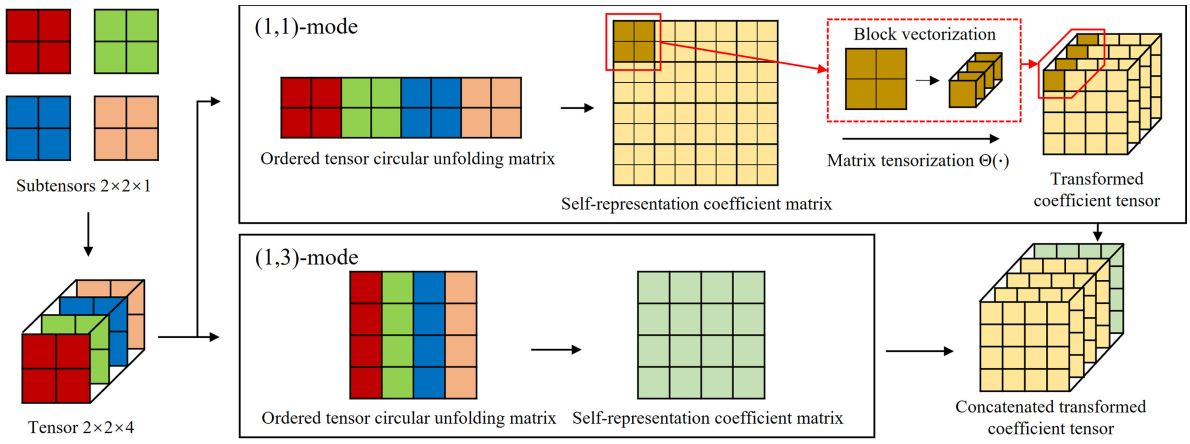

Figure 1: Generation of the transformed coefficient tensor. For the (1,1)-mode, every 2 columns of the OTCU matrix represent a subtensor. Each 2×2 block in the representation coefficient matrix is vectorized to a 4×1 vector, such that the matrix is tensorized to a transformed coefficient tensor of size $4 \times 4 \times 4$. For the (1,2)-mode, every column represents a subtensor, such that the representation matrix itself can be directly treated as a $4 \times 4 \times 1$ transformed coefficient tensor. Finally, the coefficient tensors corresponding to the different modes can be concatenated along the 3-rd dimension.

low-rank tensor space, and prove that the representative coefficient tensor will be of low tensor rank.

- We develop a multi-mode tensor space clustering algorithm for data lying in a union of tensor spaces. The algorithm finds a low-tensor-rank self-representation across different modes of unfolding matrices and imposes cluster consistency across all modes. Further, the algorithm can handle tensors with missing entries.
- We conduct experiments on both synthetic and real data demonstrating that the proposed clustering algorithm outperforms existing algorithms.

Related Work

Multi-way data clustering. Some clustering algorithms that take advantage of the intrinsic structure of multi-way data were developed in recent years. The ordered subspace clustering (OSC) algorithm (Tierney, Gao, and Guo 2014) and the ordered sparse clustering with block-diagonal prior (BD-OSC) (Wu et al. 2015) assume the data is obtained sequentially in time or space. Several algorithms model 3-order tensor data by a union of free submodules, including the sparse submodule clustering method (SSmC) (Kernfeld, Aeron, and Kilmer 2014), sparse and low-rank submodule clustering (SLRSmC) (Piao et al. 2016) and t-SVD-based tensor low-rank representation (t-SVD-TLRR) (Zhou et al. 2019). By directly enforcing low Tucker rank on the tensor data, a tensor low-rank representation (Tucker-TLRR) and a sparse coding-based (Tucker-TLRRSC) subspace clustering method were proposed in (Fu et al. 2016). Further, (Jegelka, Sra, and Banerjee 2009) developed a combination tensor clustering (CoTeC) algorithm, which clusters data along each dimension. In sharp contrast to the foregoing methods, here we assume that the data lies in a tensor space and define a new representation coefficient for tensor unfolding matrices of each mode. As this is a different formulation, it necessitates entirely different algorithm design.

Multi-view subspace clustering (MVSC). In this framework, multiple feature sets are integrated to perform clustering (Gao et al. 2015). By exploiting both the consistency and complementarity of information across multi-view data, MVSC was shown to yield better clustering performance than single view clustering, which was demonstrated in numerous recent works including (Cao et al. 2015; Zhang et al. 2017; Xie et al. 2018; Gao et al. 2020; Wang et al. 2018b). Our work shares some similarity with MVSC in that it leverages cluster consistency across different 'views' to improve performance, but also addresses a different problem and uses an altogether different methodology. In particular, unlike MVSC where each view has a different feature set, in our setting, the data from the different modes originates and is mapped from the same tensor, which calls for alternative means to capture existing correlations across the different modes to yield accurate clustering.

Subspace clustering with missing data (SCMD). Many algorithms with different underlying principles were developed to address SCMD when some of the data entries are missing (Lane et al. 2019; Yang, Robinson, and Vidal 2015; Elhamifar 2016; Fan and Chow 2017; Li and Vidal 2016; Pimentel-Alarcón et al. 2016). Incomplete multi-view clustering (IMC) extends SCMD to the multi-view domain, where the feature sets may have some missing views and missing entries. Several methods were proposed for solving the IMC problem, such as (Tao et al. 2019; Li, Jiang, and Zhou 2014; Shao, He, and Philip 2015; Xu, Tao, and Xu 2015; Zhao, Liu, and Fu 2016). Our work extends SCMD by modeling data in tensor spaces.

Preliminaries

Notation. Uppercase script letters are used to denote tensors (e.g., $\mathcal{X}$), and boldface letters to denote matrices (e.g., $\mathbf{X}$). An N-order tensor is defined as $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, where $I_i, i \in [N] := \{1, \ldots, N\}$, is the dimension

Tensor model	$r^*_{\langle l,d\rangle}$ for $\mathbf{X}_{\langle l,d\rangle}$
CP	$r(r_1 = r_2 = \cdots = r_N = r)$
Tucker	$\min\{\prod_{i=l}^{l+d-1} r_i, \prod_{j=l+d}^{l+N-1} r_j\}$
Tensor Train (TT)	$r_l r_{l+d}$
Tensor Ring (TR)	$r_l r_{l+d}$

Table 1: $r^*_{\langle l,d\rangle}$ for different tensor models.

of the i-th way of the tensor, and $\mathcal{X}_{i_1 \dots i_N}$ denotes the $(i_1, i_2, \dots, i_N)$-th entry of tensor $\mathcal{X}$. For a 3-rd order tensor (i.e., $N = 3$), the notation $\mathcal{X}(:,:,i), \mathcal{X}(:,i,:), \mathcal{X}(i,:,:)$ denotes the frontal, lateral, horizontal slices of $\mathcal{X}$, respectively. The Frobenius norm of tensor $\mathcal{X}$ is defined as $\|\mathcal{X}\|_F = \sqrt{\sum_{i_1 \dots i_N} |\mathcal{X}_{i_1 \dots i_N}|^2}$. Next, we review the definition of the tensor circular unfolding along with its properties.

Definition 1 (Tensor Circular Unfolding (TCU) (Yu et al. 2018, 2019))**.** *Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an N-order tensor. The (l,d)-mode tensor circular unfolding of $\mathcal{X}$ is a matrix denoted by $\mathbf{X}_{\langle l,d\rangle}$ of size $\prod_{i=l}^{l+d-1} I_i \times \prod_{j=l+d}^{l+N-1} I_j$, whose elements are defined by*

$$X_{\langle l,d\rangle}\left(\overline{i_l \cdots i_{l+d-1}}, \overline{i_{l+d} \cdots i_{l+N-1}}\right) = \mathcal{X}(i_1, i_2, \dots, i_N)$$

with $I_{l+N} = I_l, i_{l+N} = i_l$ for $l = 1, \dots, N$ and

$$\overline{i_a \cdots i_b} = 1 + \sum_{c=a}^{b} (i_c - 1) \prod_{j=a}^{c-1} I_j \, .$$

TCU was originally proposed for the tensor ring decomposition model (Zhao et al. 2016). In this work, we apply TCU to more tensor decomposition models, including CANDECOMP/PARAFAC (CP) (Kiers 2000), Tucker (Tucker 1966) and Tensor Train (TT) (Oseledets 2011). The following theorem characterizes the rank property of TCU under different tensor models.

Theorem 1. *Assume $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is an N-order tensor under the CP/Tucker/TT/TR decomposition model with multilinear rank $\mathbf{r} = [r_1, \dots, r_N]$. Then, for each tensor, the rank $r_{\langle l,d\rangle}$ of the circular unfolding matrix $\mathbf{X}_{\langle l,d\rangle}$ satisfies*

$$r_{\langle l,d\rangle} := \operatorname{rank}\left(\mathbf{X}_{\langle l,d\rangle}\right) \leq r^*_{\langle l,d\rangle} \, ,$$

where $r^*_{\langle l,d\rangle}$ is given in Table 1 with $r_{l+N} = r_l$.

Theorem 1 extends (Yu et al. 2019, Theorem 1) on the TR model. It can be seen from Table 1 that $r^*_{\langle l,d\rangle}$ is directly related to the tensor rank. If $r^*_{\langle l,d\rangle} \ll \max\{\prod_{i=l}^{l+d-1} I_i, \prod_{j=l+d}^{l+N-1} I_j\}$, $\mathbf{X}_{\langle l,d\rangle}$ will be low rank. In other words, the columns of $\mathbf{X}_{\langle l,d\rangle}$ can be represented as a linear combination of a basis consisting of at most $r^*_{\langle l,d\rangle}$ columns from $\mathbf{X}_{\langle l,d\rangle}$. Based on this observation, we define a tensor space containing a set of subtensors.

Definition 2 (Tensor space)**.** *Given a set of N-order subtensors $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_{N-1} \times 1}, i = 1, \dots, I_N$, define $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ as the concatenation of all subtensors along the N-th dimension. If $\mathcal{X}$ has multilinear rank $\mathbf{r} = [r_1, \dots, r_N]$ under the CP/Tucker/TT/TR decomposition model, we say that the subtensors $\{\mathcal{X}_i\}_{i=1}^N$ lie in a CP/-Tucker/TT/TR tensor space with dimension (rank) vector $\mathbf{r}$. Specifically, the columns of the TCU matrix $\mathbf{X}_{\langle l,d\rangle}$ for any $l, d \in [1, \dots, N]$ lie in an $r_{\langle l,d\rangle}$-dimensional subspace.*

We remark that our definition of tensor space above is more general than that in (Ashraphijuo and Wang 2020) and (Zhou et al. 2019) where the unfolding dimension and the tensor order are restricted to N and 3, respectively.

Tensor Space Clustering

Similar to subspace clustering, the task of tensor space clustering is to group subtensors drawn from a union of tensor spaces, which can be stated as follows.

Data Model 1 (Tensor space clustering (Ashraphijuo and Wang 2020))**.** *The data tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ can be represented as $\mathcal{X} = \Phi(\mathcal{X}^1, \dots, \mathcal{X}^k, \dots, \mathcal{X}^K)$, where $\mathcal{X}^k \in \mathbb{R}^{I_1 \times I_2 \times \dots I_{N-1} \times c_k}, k = 1, 2, \dots, K$, $I_N = \sum_{k=1}^K c_k$. The operator $\Phi(\cdot)$ permutes and concatenates the subtensors $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_{N-1} \times 1}, i = 1, \dots, I_N$ along the N-th dimension. The subtensors in $\mathcal{X}^k$ lie in the k-th tensor space of dimension vector $\mathbf{r}^k = [r^k_1, \dots, r^k_N]$.*

In this work, we are interested in clustering the subtensors $\mathcal{X}_i$ into K groups such that the subtensors in each group belong to the same tensor space.

Low-Tensor-Rank Representation

Low Rank Representation (LRR)

The LRR algorithm (Liu et al. 2010) finds a low-rank representation of a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ whose columns are drawn from a union of K independent subspaces of dimensions $\{r_k\}_{k=1}^K$. It finds a representation coefficient matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ by solving the convex optimization problem

$$\min_{\mathbf{C}} \|\mathbf{C}\|_* \quad \text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{C} \, , \tag{1}$$

where $\|.\|_*$ denotes the nuclear norm. A segmentation of the data is then obtained by applying spectral clustering to an affinity matrix $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}^T|$. The underlying principle of LRR is that if the rank $r = \operatorname{rank}(\mathbf{X}) = \sum_{k=1}^K r_k$ of the data matrix $\mathbf{X}$ is such that $r \ll \min\{m, n\}$, then the representation coefficient matrix $\mathbf{C}$ will be low rank.

In the tensor space clustering problem, given a tensor satisfying Data Model 1, for $\mathcal{X}^k$ in the k-th tensor space, we have that $\operatorname{rank}(\mathbf{X}^k_{\langle l,d\rangle}) = r^k_{\langle l,d\rangle}$, i.e., the columns of $\mathbf{X}^k_{\langle l,d\rangle}$ lie in an $r^k_{\langle l,d\rangle}$-dimensional subspace. Therefore, for a tensor $\mathcal{X}$ lying in a union of K tensor spaces, the columns of the unfolding matrix $\mathbf{X}_{\langle l,d\rangle}$ lie in a union of K subspaces of dimensions $\{r^k_{\langle l,d\rangle}\}_{k=1}^K$. Hence, we can apply the LRR subspace clustering algorithm to $\mathbf{X}_{\langle l,d\rangle}$. Specifically, the representation matrix can be obtained by solving

$$\min_{\mathbf{C}} \|\mathbf{C}\|_* \quad \text{s.t.} \quad \mathbf{X}_{\langle l,d\rangle} = \mathbf{X}_{\langle l,d\rangle}\mathbf{C} \, , \tag{2}$$

and the columns can then be grouped via spectral clustering. However, note that multiple columns in $\mathbf{X}_{\langle l,d\rangle}$ may represent the same subtensor (see Figure 1), but $\|\mathbf{C}\|_*$ does not

capture this 'group' consistency. To this end, one should use an alternative constraint on $\mathbf{C}$, which will be introduced in this work.

Low-Tensor-Rank Representation (LTRR)

In this paper, we develop a new representation for tensor space clustering. We first propose an ordered TCU to group the columns belonging to the same subtensor.

Definition 3 (Ordered Tensor Circular Unfolding (OTCU))**.** *Given a TCU matrix $\mathbf{X}_{\langle l,d\rangle}$ of tensor $\mathcal{X}$, the ordered TCU $\bar{\mathbf{X}}_{\langle l,d\rangle}$ is obtained by re-sorting the columns of $\mathbf{X}_{\langle l,d\rangle}$ as*

$$\bar{\mathbf{X}}_{\langle l,d\rangle:,v} = \begin{cases} \left[\mathbf{X}_{\langle l,d\rangle}\right]_{:,o}, N-l+1 > d \\ \left[\mathbf{X}^T_{\langle l,d\rangle}\right]_{:,o}, N-l+1 \leq d \end{cases} \tag{3}$$

where $\mathbf{X}_{:,i}$ and $\mathbf{X}_{i,:}$ represents the i-th column and row of the matrix $\mathbf{X}$, respectively, The relation between v and o is $o = D((p(v)-1)I_N + c(v) - 1) + q(v)$, where

$$c(v) = \lfloor (v-1)/(LD) \rfloor + 1, q(v) = \mathrm{mod}\{v-1, D\} + 1$$
$$p(v) = \lfloor (v - (c(v)-1)LD - 1)/D \rfloor + 1$$
$$\{L, D\} = \begin{cases} \{\prod_{i=1}^{l-1} I_i, \prod_{j=l+d}^{N-1} I_j\}, N-l+1 > d \\ \{\prod_{i=1}^{k+d-N-1} I_i, \prod_{j=l}^{N-1} I_j\}, N-l+1 \leq d \end{cases}$$

Specifically, by rewriting $\bar{\mathbf{X}}_{\langle l,d\rangle}$ as the group form $\bar{\mathbf{X}}_{\langle l,d\rangle} = [\mathbf{X}^1\ \mathbf{X}^2\ \ldots\ \mathbf{X}^{I_N}]$, $\mathbf{X}^i \in \mathbb{R}^{[\prod_{i=1}^{N-1} I_i/(LD)] \times LD}$ is a submatrix unfolded from the i-th subtensor.

Figure 1 illustrates the process of generating the transformed coefficient tensor. Using the OTCU operation above, the columns representing the same subtensors are grouped. After obtaining the representation matrix $\mathbf{C}$, each $LD \times LD$ block is vectorized into an L^2D^2 vector. Finally, all the vectors are stacked in a 3-rd order tensor of size $I_N \times I_N \times L^2D^2$. We denote the resulting transformed coefficient tensor as $\mathcal{C} = \Theta(\mathbf{C})$, where $\Theta(\cdot)$ denotes the above tensorization procedure, whereby $\mathbf{C}$ is reshaped into a tensor $\mathcal{C} \in \mathbb{R}^{I_N \times I_N \times L^2D^2}$. The following lemma characterizes the tensor $\mathcal{C}$ obtained from a tensor $\mathcal{X}$ lying in a tensor space.

Lemma 1. *Suppose $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ is an N-order tensor with multilinear rank $[r_1, \ldots r_N]$. For an OTCU matrix $\bar{\mathbf{X}}_{\langle l,d\rangle}$, let $\mathbf{C}$ be the self-representation matrix obtained using (2). Then, the transformed coefficient tensor $\mathcal{C} = \Theta(\mathbf{C}) \in \mathbb{R}^{I_N \times I_N \times L^2D^2}$ is a tensor with multilinear rank $[r_{<1,N-1>}, 1, r_{<1,N-1>}]$ under the TR model. Further, the concatenation of the tensor $\mathcal{C}$ obtained from $\bar{\mathbf{X}}_{\langle l,d\rangle}$ for any l, d along the 3-rd dimension will also have the multilinear rank $[r_{<1,N-1>}, 1, r_{<1,N-1>}]$.*

The following proposition generalizes Lemma 1 for a tensor generated from a union of tensor spaces.

Proposition 1. *Suppose $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ is an N-order tensor generated from Data model 1, $\mathbf{C}_{l,d}$ is a self-representation matrix for the OTCU matrix $\bar{\mathbf{X}}_{\langle l,d\rangle}$ obtained using (2), $l, d = 1, \ldots, N$, and $\mathcal{C}_{l,d}$ is the corresponding transformed coefficient tensor. Then, for a tensor $\mathcal{C}$ generated by concatenating any tensors in $\{\mathcal{C}_{l,d}\}_{l,d=1}^{N}$ along the 3-rd dimension, the multilinear rank of tensor $\mathcal{C}$ will be $[\sum_{k=1}^{K} r^k_{<1,N-1>}, 1, \sum_{k=1}^{K}(r^k_{<1,N-1>})]$ under TR model.*

We name the above representation for OTCU the low-tensor-rank representation (LTRR). LTRR has two main advantages. First, it allows us to capture various group consistency relations among the columns that would not be possible to capture by the rank constraint on $\mathbf{C}$ in LRR. Second, it resolves the problem of integrating representation matrices for different unfolding matrices of disparate dimensions, making it possible to ensure consistency across the different modes (see supplementary document for more explanation).

Proposed Multi-Mode Clustering Method

In this section, we present our algorithm for tensor space clustering based on LTRR. Similar to multi-view clustering (Gao et al. 2020), we treat an (l, d)-mode OTCU matrix $\bar{\mathbf{X}}_{\langle l,d\rangle}$ as a view and find a representation matrix $\mathbf{C}_{l,d}$ for $\bar{\mathbf{X}}_{\langle l,d\rangle}$. The rank constraint is imposed on a representation tensor $\mathcal{C}$ integrated from different representation matrices. Specifically, given M modes $\{l_i, d_i\}_{i=1}^{M}$, we solve the following optimization problem

$$\begin{aligned} &\min_{\mathbf{E},\mathcal{C}} \lambda \|\mathbf{\Omega} \circ \mathbf{E}\|_{2,1} + \mathrm{rank}_{\mathrm{tr}}(\mathcal{C}) \quad \text{s.t.} \\ &\mathbf{X}_m = \mathbf{X}_m\mathbf{C}_m + \mathbf{E}_m, m = 1, \ldots, M \\ &\mathcal{C} = \Psi\left(\Theta(\mathbf{C}_1), \Theta(\mathbf{C}_2), \ldots, \Theta(\mathbf{C}_M)\right) \\ &\mathbf{E} = [f(\mathbf{E}_1); f(\mathbf{E}_2); \ldots; f(\mathbf{E}_M)] \end{aligned} \tag{4}$$

where $\mathbf{X}_m$ is short for $\bar{\mathbf{X}}_{\langle l_m,d_m\rangle}$, $\mathrm{rank}_{\mathrm{tr}}(\mathcal{C})$ denotes the tensor ring rank of $\mathcal{C}$, $\circ$ is the Hadamard product, and the $\Psi(\cdot)$ operator stacks the tensors along the 3-rd dimension. $f(\mathbf{E}_m)$ is obatined by first folding the self-expressiveness error matrix for the m-th mode, $\mathbf{E}_m$, back to a tensor with the same size as $\mathcal{X}$, then unfolding along the $(N-1)$-th dimension. The $L_{2,1}$-norm $\|\cdot\|_{2,1}$ is used to promote column sparsity of $\mathbf{E}$ (Zhang et al. 2017). We also define the inverse mapping $\mathbf{C}_m = \Psi_m^{-1}(\mathcal{C})$.

We also consider the case where $\mathcal{X}$ has missing entries. In this case, we fill the missing entries with 0 and introduce a mask matrix $\mathbf{\Omega}$. An entry of $\mathbf{\Omega}$ is 1 if the corresponding error is computed from observed entries, otherwise it is set to 0. The intuition is similar to the projected zero-filled SSC (PZF-SSC) algorithm for subspace clustering with missing data (Lane et al. 2019), where the self-expressiveness error is only measured on the observed entries to discount unwarranted errors over the zero-filled unobserved entries.

The formulation above resembles the t-SVD based Multi-view Subspace Clustering (t-SVD-MSC) (Xie et al. 2018) but there are important differences. First, here the unfolding matrices from all views share the same data. Second, the representation matrices $\mathbf{C}_l$ may not have the same size so they cannot be simply merged. Third, t-SVD-MSC does not consider the missing data case. Fourth, we use a tensor ring rank model for the constraint on $\mathcal{C}$.

Solving the Problem Using Tensor-Ring Nuclear Norm Minimization and ADMM

Minimizing $\mathrm{rank}_{\mathrm{tr}}(\mathcal{C})$ is an NP-hard problem. In this work, we use a tensor ring nuclear norm minimization-based

method (Yu et al. 2019) to relax the tensor ring rank constraint to a convex optimization problem. Specifically, by introducing new variables $\{\mathcal{G}^{(i)}\}_{i=1}^3$, (4) can be rewritten as

$$\begin{aligned}
&\min_{\mathbf{E},\mathcal{C}} \lambda\|\mathbf{\Omega}\circ\mathbf{E}\|_{2,1}+\sum\nolimits_{i=1}^{3}\frac{1}{3}\|\mathbf{G}^{(i)}_{<i,2>}\|_* \quad \text{s.t.}\\
&\mathbf{X}_m=\mathbf{X}_m\mathbf{C}_m+\mathbf{E}_m, m=1,\dots,M,\\
&\mathcal{C}=\Psi\left(\Theta(\mathbf{C}_1),\Theta(\mathbf{C}_2),\dots,\Theta(\mathbf{C}_M)\right)\\
&\mathbf{E}=[f(\mathbf{E}_1);f(\mathbf{E}_2);\dots;f(\mathbf{E}_M)]\\
&\mathcal{G}^{(i)}=\mathcal{C}, i=1,2,3\,.
\end{aligned} \tag{5}$$

The problem in (5) can be efficiently solved using an ADMM-based algorithm. Specifically, by introducing new variables $\{\mathcal{G}^{(i)}\}_{i=1}^3$, $\{\mathcal{W}^{(i)}\}_{i=1}^3$ and $\{\mathbf{Y}_m\}_{m=1}^M$, (5) can be reformulated as the unconstrained minimization of

$$\begin{aligned}
&\mathcal{L}\left(\mathcal{C};\mathbf{E};\mathcal{G}^{(1)},\dots,\mathcal{G}^{(3)};\mathcal{W}^{(1)},\dots,\mathcal{W}^{(3)};\mathbf{Y}_1,\dots\mathbf{Y}_M\right)\\
&=\lambda\|\mathbf{\Omega}\circ\mathbf{E}\|_{2,1}+\sum\nolimits_{i=1}^{3}\frac{1}{3}\|\mathbf{G}^{(i)}_{<i,2>}\|_*\\
&+\sum_{l=m}^{M}\left(\langle\mathbf{Y}_m,\mathbf{X}_m-\mathbf{X}_m\mathbf{C}_m-\mathbf{E}_m\rangle+\frac{\mu}{2}\|\mathbf{X}_m-\mathbf{X}_m\mathbf{C}_m-\mathbf{E}_m\|_F^2\right)\\
&+\sum\nolimits_{i=1}^{3}\left(\langle\mathcal{W}^{(i)},\mathcal{C}-\mathcal{G}^{(i)}\rangle+\frac{\beta}{2}\left\|\mathcal{C}-\mathcal{G}^{(i)}\right\|_F^2\right)
\end{aligned} \tag{6}$$

Then, we alternatively update the variables as follows.

1. **Update** $\mathbf{C}_m$: The representation coefficient matrix $\mathbf{C}_m$ for the m-th mode can be obtained as

$$\begin{aligned}
\mathbf{C}_m=\arg\min_{\mathbf{C}}\mu\left\|\mathbf{X}_m-\mathbf{X}_m\mathbf{C}-\mathbf{E}_m+\frac{1}{\mu}\mathbf{Y}_m\right\|_F^2\\
+\beta\sum\nolimits_{i=1}^{3}\left\|\mathbf{C}-\mathbf{G}_m^{(i)}+\frac{1}{\beta}\mathbf{W}_m^{(i)}\right\|_F^2
\end{aligned} \tag{7}$$

where $\mathbf{G}_m^{(i)}$ and $\mathbf{W}_m^{(i)}$ are obtained by $\Psi_m^{-1}(\mathcal{G}^{(i)})$ and $\Psi_m^{-1}(\mathcal{W}^{(i)})$, respectively. The above problem has a closed-form solution

$$\begin{aligned}
\mathbf{C}_m=&\left(\frac{3\beta}{\mu}\mathbf{I}+\mathbf{X}_l^T\mathbf{X}_m\right)^{-1}\left(\mathbf{X}_m^T\mathbf{X}_m-\mathbf{X}_m^T\mathbf{E}_m\right.\\
&\left.+\frac{1}{\mu}(\mathbf{X}_m^T\mathbf{Y}_m+\beta\sum\nolimits_{i=1}^{3}\mathbf{G}_m^{(i)}-\sum\nolimits_{i=1}^{3}\mathbf{W}_m^{(i)})\right)
\end{aligned} \tag{8}$$

2. **Update** $\mathbf{E}$: The variable $\mathbf{E}$ can be updated as

$$\begin{aligned}
\mathbf{E}=&\arg\min_{\mathbf{E}}\lambda\|\mathbf{\Omega}\circ\mathbf{E}\|_{2,1}+\sum\nolimits_{m=1}^{M}(\langle\mathbf{Y}_m,\mathbf{X}_m-\mathbf{X}_m\mathbf{C}_m\\
&-\mathbf{E}_m\rangle+\frac{\mu}{2}\left\|\mathbf{X}_m-\mathbf{X}_m\mathbf{C}_m-\mathbf{E}_m\right\|_F^2)\\
=&\arg\min_{\mathbf{E}}\frac{\lambda}{\mu}\|\mathbf{\Omega}\circ\mathbf{E}\|_{2,1}+\frac{1}{2}\|\mathbf{E}-\mathbf{B}\|_F^2
\end{aligned} \tag{9}$$

where $\mathbf{B}$ is constructed by vertically concatenating the matrices $\mathbf{X}_m-\mathbf{X}_m\mathbf{C}_m+\frac{1}{\mu}\mathbf{Y}_m$ together along the columns. The above problem has the following solution (Xie et al. 2018)

$$\mathbf{E}_{:,i}=\begin{cases}\frac{\|\mathbf{\Omega}_{:,i}\circ\mathbf{B}_{:,i}\|_2-\frac{\lambda}{\mu}}{\|\mathbf{\Omega}_{:,i}\circ\mathbf{B}_{:,i}\|_2}\mathbf{\Omega}_{:,i}\circ\mathbf{B}_{:,i}+(1-\mathbf{\Omega}_{:,i})\circ\mathbf{B}_{:,i}, & \\ \qquad\qquad\qquad\qquad\|\mathbf{\Omega}_{:,i}\circ\mathbf{B}_{:,i}\|_2>\frac{\lambda}{\mu}\\ (1-\mathbf{\Omega}_{:,i})\circ\mathbf{B}_{:,i}, \qquad\text{otherwise}\end{cases} \tag{10}$$

Algorithm 1: Multi-mode tensor space clustering (MMTSC)

Input: $\mathcal{X}$, $\mathbf{\Omega}$, $\{(l_m,d_m)\}_{m=1}^M$, β, λ, μ, ρ and ϵ.

initial matrices $\{\mathbf{C}_m\}_{m=1}^M$, $\{\mathbf{Y}_m\}_{m=1}^M$, tensors $\{\mathcal{G}^{(i)}\}_{i=1}^3$ and $\{\mathcal{W}^{(i)}\}_{i=1}^3$.
repeat
  compute $\{\mathbf{C}_m\}_{m=1}^M$ using (8).
  compute $\mathbf{E}$ using (10).
  compute $\{\mathcal{G}^{(i)}\}_{i=1}^3$ using (12).
  compute $\{\mathbf{Y}_m\}_{m=1}^M$ and $\{\mathcal{W}^{(i)}\}_{i=1}^3$ using (13).
  update $\mu=\rho\mu$, $\beta=\rho\beta$.
until $\sum_{i=1}^3\|\mathcal{C}-\mathcal{G}^{(i)}\|_F^2/\|\mathcal{C}\|_F^2<\epsilon$
obtain the affinity matrix by $\mathbf{A}=\sum_{i=1}^S|\mathcal{C}(:,:,i)|+|\mathcal{C}(:,:,i)|^T$, where S is the number of frontal slices.
apply spectral clustering using $\mathbf{A}$ and obtain cluster label vector $\mathbf{c}$.

Output: clustering label $\mathbf{c}$

3. **Update** $\mathcal{G}^{(i)}$: The tensors $\mathcal{G}^{(i)}$ can be obtained by solving

$$\mathcal{G}^{(i)}=\arg\min_{\mathcal{G}}\|\mathbf{G}_{<i,2>}\|_*+3\beta\|\mathcal{C}-\mathcal{G}+\frac{1}{\beta}\mathcal{W}^{(i)}\|_F^2 \tag{11}$$

The above problem has a closed-form solution

$$\mathcal{G}^{(i)}=\text{fold}_i\left[\mathbf{\Gamma}_{\frac{1}{3\beta}}\left(\mathbf{M}_{<i,2>}\right)\right] \tag{12}$$

where $\mathcal{M}=\mathcal{C}-\frac{1}{\beta}\mathcal{W}^{(i)}$, $\mathbf{\Gamma}(\cdot)$ denotes the thresholding SVD operation and $\text{fold}_i(\cdot)$ folds the matrix back to the tensor.

4.**Update** $\{\mathcal{W}^{(i)}\}_{i=1}^3$ **and** $\{\mathbf{Y}_m\}_{m=1}^M$:

$$\begin{aligned}
\mathbf{Y}_m&=\mathbf{Y}_m+\mu(\mathbf{X}_m-\mathbf{X}_m\mathbf{C}_m-\mathbf{E}_m),\\
\mathcal{W}^{(i)}&=\mathcal{W}^{(i)}+\beta(\mathcal{C}-\mathcal{G}^{(i)})
\end{aligned} \tag{13}$$

We name the above algorithm Multi-mode tensor space clustering (MMTSC). The pseudocode for MMTSC is presented in Algorithm 1. To improve convergence, we set the parameters μ and β to increase geometrically with the iterations (Nishihara et al. 2015).

Proving convergence in multi-block settings is known to be generally hard. We note that the authors in (Xie et al. 2018) established convergence guarantees of the t-SVD-MSC under two sufficient conditions, namely, full column rank of representations of the unfolding matrices and monotonicity of the iterates by the convexity of the Lagrangian function. Despite the differences in formulations between our algorithm and t-SVD-MSC, we could still establish similar convergence guarantees for the proposed algorithm under the same conditions of t-SVD-MSC.

Experimental Results

In this section, we carry out experiments to verify the performance of the proposed method. We compare the performance with several existing clustering methods, including matrix-based methods LRR, SSC, SSC-PZF (Yang, Robinson, and Vidal 2015), the sequential OSC method (Tierney, Gao, and Guo 2014), and the tensor-based method t-SVD-TLRR (Zhou et al. 2019). We use the clustering accuracy (percentage of correctly classified subtensors) for

evaluation with synthetic data. For real data, the performance is evaluated using the accuracy and the normalized mutual information (NMI) between the clustering results and the ground truth labels. The suffix "+TC" appended to a method's name designates first applying the tensor ring completion algorithm PTRC (Yu et al. 2020) to complete the missing pixels then applying the clustering method. For the proposed MMTSC, we add a suffix "-M" to represent the number of modes used by the algorithm. We set $l_m = m, d_m = \lceil N/2 \rceil, m = 1, \ldots, M$. The parameters for MMTSC are set to $\lambda = 0.1M$, $\mu = 0.01$, $\beta = 0.1$, $\rho = 1.1$, $\epsilon = 10^{-3}$. We tune the parameters of the algorithms that we compare against so that they achieve their best performance in our experiments. All experiments were performed using MATLAB R2018b on a desktop PC with a 2.6-GHz processor and 16GB of RAM. The code is available at https://github.com/he1c/LTRR-TensorSC.

Synthetic Data

In this section, the performance of the algorithms is investigated using synthetic data. The data is generated from a union of TR tensor spaces following Data Model 1, with model parameters $N = 4$, $K = 10$, $I_1 = \cdots = I_3 = 10$. The elements of TR rank $\mathbf{r}^k$ for all k are set to the same value r. All values of c_k are set to the same value $c = 10$. The entries of the TR core tensors are generated independently from a Gaussian distribution with zero mean and unit variance. All tensors are scaled to the range $[-1, 1]$. Then, $p \times 100\%$ of entries are selected as the observed entries and the remaining entries are marked as unobserved (filled with 0). Further, i.i.d. additive Gaussian noise with zero mean and variance 0.04 is added to the observed entries of the tensor.

First, we analyze the clustering performance with different rank r and observation rate p. The values of r and p are selected from the intervals $[1, 20]$ and $[0.05, 1]$, respectively. For a fixed combination of r and p, the clustering accuracy is evaluated by averaging over 20 Monte Carlo (MC) runs with different tensors, observation locations and noise. Figure 2 displays the phase transition of the accuracy as function of r and p for different clustering algorithms, where whiter regions indicate higher accuracy. Our algorithm is shown to exhibit better phase transitions. Specifically, given a fixed r,

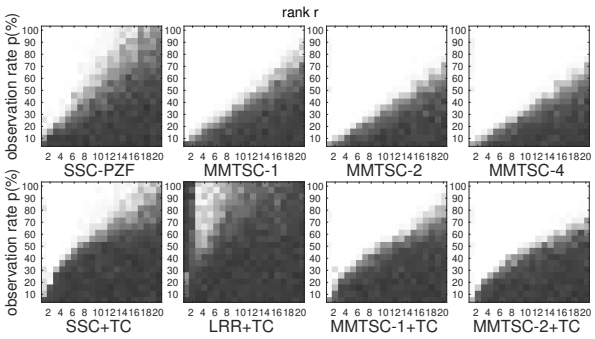

Figure 2: Phase transitions for different rank r and observation rate $p(\%)$.

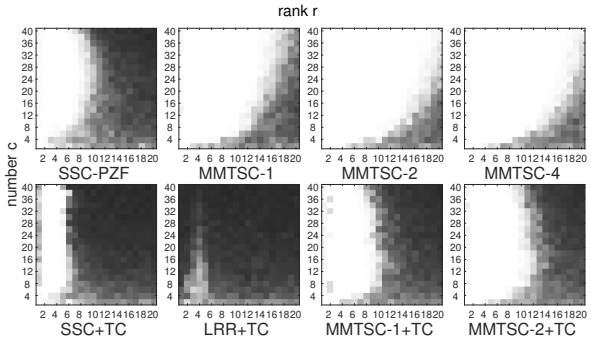

Figure 3: Phase transitions for different rank r and number of subtensors per cluster c ($p = 0.5$).

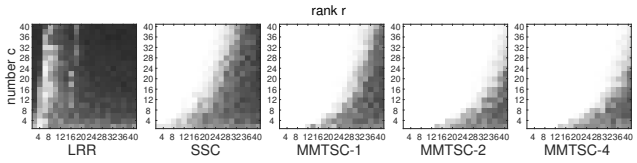

Figure 4: Phase transitions for different rank r and number of subtensors per cluster c ($p = 1$).

MMTSC with 2 and 4 modes requires a smaller observation rate to achieve accurate clustering.

Second, we study the clustering performance with different numbers of subtensors per cluster. The observation rate p is selected as 0.5 and 1. The number of subtensors per cluster c is varied from 2 to 40. For each c, 20 MC runs are performed. The phase transition plots of the average accuracy in the c and r plane are depicted in Figures 3 and 4 for different algorithms. As shown, MMTSC with 2 and 4 modes has the best phase transitions. Specifically, given a certain r, it requires less subtensors per cluster for accurate clustering. Further, as can be seen from Figure 3, tensor completion may bias the clustering results as it may not accurately complete the tensor due to high rank r.

Real Dataset

In this section, we verify the performance in clustering image and video sequences using three datasets described next.

YUV dataset[1] contains a collection of color videos from different scenes. For each video, a sequence consisting of the first 50 frames is selected. Each frame is converted to grayscale and down-sampled to 36×44, yielding a subtensor of size $36 \times 44 \times 50$. For each MC run, we randomly select 10 videos clips. The video clips (subtensors) are concatenated along the temporal dimension resulting in a tensor of size $36 \times 44 \times 800$. For the proposed method, the observed tensor is reshaped to a 5-order tensor of size $6 \times 6 \times 4 \times 11 \times 800$, while for the matrix-based methods the observed tensor is reshaped to a matrix of size 1584×800.

UCSD dataset[2] provides 18 short videos containing differ-

[1] http://trace.eas.asu.edu/yuv/

[2] http://www.svcl.ucsd.edu/projects/background_subtraction/

Dataset	p	metric	LRR+TC	SSC+TC	OSC+TC	t-SVD-TLRR+TC	SSC-PZF	MMTSC-1	MMTSC-1+TC	MMTSC-2	MMTSC-2+TC
YUV	0.1	Acc	0.1680	0.3850	0.1800	0.3980	0.6780	0.7260	0.6920	0.8820	**0.9010**
		NMI	0.0433	0.2785	0.0605	0.3309	0.7094	0.6724	0.6740	0.8330	**0.8920**
	0.3	Acc	0.1790	0.9750	0.7690	0.9950	0.8460	0.9630	**1**	0.9800	**1**
		NMI	0.0541	0.9820	0.7375	0.9906	0.9330	0.9593	**1**	0.9760	**1**
	0.5	Acc	0.1630	0.8550	0.7760	**1**	0.9090	0.9520	**1**	0.9990	**1**
		NMI	0.0428	0.9543	0.7724	**1**	0.9598	0.9583	**1**	0.9979	**1**
	1	Acc	0.1610	0.8620	0.7600	**1**	0.8530	**1**	**1**	0.9060	0.9060
		NMI	0.0347	0.9548	0.7646	**1**	0.9461	**1**	**1**	0.9606	0.9606
Coil20	0.1	Acc	0.2074	0.2009	0.2653	0.2894	0.6264	0.6296	0.5981	**0.6787**	0.6481
		NMI	0.1030	0.0857	0.1602	0.1980	0.6301	0.6250	0.5985	**0.7005**	0.6919
	0.3	Acc	0.1903	0.5597	0.7227	0.7514	0.6981	**0.8227**	0.7870	0.6894	0.7722
		NMI	0.0682	0.6416	0.6785	0.7489	0.7406	**0.8365**	0.8021	0.7527	0.8003
	0.5	Acc	0.1565	0.5597	0.7079	0.7667	0.7366	**0.8125**	0.7690	0.7463	0.7565
		NMI	0.0338	0.7348	0.7181	0.7868	0.7516	**0.8325**	0.8105	0.7763	0.8172
	1	Acc	0.1574	0.7523	0.7060	**0.7907**	0.7512	0.7537	0.7537	0.7097	0.7088
		NMI	0.0323	0.8016	0.7148	0.8225	0.8101	**0.8234**	**0.8234**	0.8096	0.8096
UCSD	0.1	Acc	0.1676	0.4810	0.1713	0.3405	0.5389	0.5320	0.7271	0.6311	**0.8172**
		NMI	0.0463	0.4154	0.0517	0.2378	0.4687	0.4912	0.6949	0.5993	**0.8267**
	0.3	Acc	0.1627	0.8161	0.3234	0.8522	0.8155	0.8763	0.8843	0.8547	**0.9126**
		NMI	0.0454	0.9218	0.2140	0.8857	0.8624	0.8980	0.9105	0.8949	**0.9590**
	0.5	Acc	0.1764	0.7862	0.4001	0.8775	0.8705	0.8837	0.9081	0.8717	**0.9228**
		NMI	0.0495	0.9116	0.2908	0.9183	0.9255	0.9200	0.9357	0.9161	**0.9634**
	1	Acc	0.1680	0.8012	0.3455	0.8979	0.7993	0.9032	0.9032	**0.9243**	**0.9243**
		NMI	0.0430	0.8911	0.2353	0.9353	0.8856	0.9425	0.9425	**0.9650**	**0.9650**

Table 2: Comparison of the clustering performance of different algorithms for three datasets.

ent dynamic scenes. For each MC run, 10 videos consisting of the first 50 frames are randomly selected (if a video has less than 50 frames, we use all its frames). The frames are resized to form a tensor of size $36 \times 56 \times N_f$ with total number of frames $N_f \leq 500$. For the proposed method, the observed tensor is reshaped to a 5-order tensor of size $6 \times 6 \times 7 \times 8 \times N_f$, while for the matrix-based methods the observed tensor is reshaped to a matrix of size $2016 \times N_f$.

Coil20 dataset[3] is a collection of grayscale images, including 20 objects taken from different angles. An image is taken for each object every 5 degrees, so each object has 72 images. For each MC run, we stack the 720 images from 10 randomly selected objects to form a tensor of size $36 \times 36 \times 720$. For the proposed method, the observed tensor is reshaped to a 5-order tensor of size $6 \times 6 \times 6 \times 6 \times 720$, while for the matrix-based methods the observed tensor is reshaped to a matrix of size 1296×720.

After obtaining the above tensor data, the subtensors (frames) are randomly shuffled along the temporal dimension. Given observation rate p, the observed tensor with missing and noisy entries is generated by removing $(1 - p) \times 100\%$ of the pixels (filled with 0), then adding i.i.d. Gaussian noise with variance 0.001 to the observed pixels.

We test the clustering performance for different values of p. For each p, 20 MC runs with different videos, frame orders, missing locations and noise are performed. Table 2 shows the average accuracy and NMI for different values of p for various clustering algorithms. The best accuracy or NMI is shown in bold and the second best is underlined. Our algorithm achieves the best overall performance. Specifically, when p is small (i.e., $p = 0.1$), the proposed method with 2 modes obtains better results than with single mode. When p becomes large, the proposed MMTSC with the single mode yields the best results in YUV and Coil20, while MMTSC with 2 modes performs best in UCSD. In general, our algorithm yields better clustering performance because our proposed LTRR representation can identify and capture more intrinsic structure from the high-order tensor.

[3]https://cs.columbia.edu/CAVE/software/softlib/coil-20.php

Conclusion

In this work we developed a novel low-tensor-rank representation (LTRR) for unfolded matrices of tensor data lying in a low-rank tensor space. The representation coefficient matrix for each unfolding matrix is tensorized to a transformed coefficient tensor, which is theoretically proved to have a low-tensor-rank property. Further, we develop a multi-mode tensor space clustering algorithm (MMTSC) that can deal with tensor space clustering with or without missing entries. The proposed method imposed a low-tensor-ring-rank constraint on the concatenated transformed coefficient tensor from different modes, which simultaneously captures the low rank property within each tensor space and maintains cluster consistency across different modes. Experimental results on both synthetic and real datasets verify the superior performance of the proposed MMTSC over existing clustering algorithms.

Acknowledgements

This work was supported by NSF CAREER Award CCF-1552497 and NSF Award CCF-2106339.

References

Ashraphijuo, M.; and Wang, X. 2020. Union of Low-Rank Tensor Spaces: Clustering and Completion. *Journal of Machine Learning Research*, 21(69): 1–36.

Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 586–594.

Elhamifar, E. 2016. High-rank matrix completion and clustering under self-expressive models. *Advances in Neural Information Processing Systems*, 29: 73–81.

Elhamifar, E.; and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11): 2765–2781.

Fan, J.; and Chow, T. W. 2017. Matrix completion by least-square, low-rank, and sparse self-representations. *Pattern Recognition*, 71: 290–305.

Fu, Y.; Gao, J.; Tien, D.; Lin, Z.; and Hong, X. 2016. Tensor LRR and sparse coding-based subspace clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 27(10): 2120–2133.

Gao, H.; Nie, F.; Li, X.; and Huang, H. 2015. Multi-view subspace clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, 4238–4246.

Gao, Q.; Xia, W.; Wan, Z.; Xie, D.; and Zhang, P. 2020. Tensor-SVD based graph learning for multi-view subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3930–3937.

Hong, W.; Wright, J.; Huang, K.; and Ma, Y. 2006. Multi-scale hybrid linear models for lossy image representation. *IEEE Transactions on Image Processing*, 15(12): 3655–3671.

Jegelka, S.; Sra, S.; and Banerjee, A. 2009. Approximation algorithms for tensor clustering. In *International Conference on Algorithmic Learning Theory*, 368–383. Springer.

Kernfeld, E.; Aeron, S.; and Kilmer, M. 2014. Clustering multi-way data: A novel algebraic approach. *arXiv preprint arXiv:1412.7056*.

Kiers, H. A. 2000. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3): 105–122.

Lane, C.; Boger, R.; You, C.; Tsakiris, M.; Haeffele, B.; and Vidal, R. 2019. Classifying and comparing approaches to subspace clustering with missing data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.

Li, C.-G.; and Vidal, R. 2016. A structured sparse plus structured low-rank framework for subspace clustering and completion. *IEEE Transactions on Signal Processing*, 64(24): 6557–6570.

Li, S.-Y.; Jiang, Y.; and Zhou, Z.-H. 2014. Partial multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Liu, G.; Lin, Z.; Yu, Y.; et al. 2010. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning (ICML)*, volume 1, 8.

Lu, C.-Y.; Min, H.; Zhao, Z.-Q.; Zhu, L.; Huang, D.-S.; and Yan, S. 2012. Robust and efficient subspace segmentation via least squares regression. In *European Conference on Computer Vision*, 347–360. Springer.

Nishihara, R.; Lessard, L.; Recht, B.; Packard, A.; and Jordan, M. 2015. A general analysis of the convergence of ADMM. In *International Conference on Machine Learning*, 343–352.

Oseledets, I. V. 2011. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5): 2295–2317.

Parsons, L.; Haque, E.; and Liu, H. 2004. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1): 90–105.

Peignier, S.; Rigotti, C.; and Beslon, G. 2015. Subspace clustering using evolvable genome structure. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 575–582.

Piao, X.; Hu, Y.; Gao, J.; Sun, Y.; and Lin, Z. 2016. A submodule clustering method for multi-way data by sparse and low-rank representation. *arXiv preprint arXiv:1601.00149*.

Pimentel-Alarcón, D.; Balzano, L.; Marcia, R.; Nowak, R.; and Willett, R. 2016. Group-sparse subspace clustering with missing data. In *IEEE Statistical Signal Processing Workshop (SSP)*, 1–5.

Shao, W.; He, L.; and Philip, S. Y. 2015. Multiple incomplete views clustering via weighted nonnegative matrix factorization with $L_{2,1}$ regularization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 318–334. Springer.

Tao, H.; Hou, C.; Yi, D.; Zhu, J.; and Hu, D. 2019. Joint embedding learning and low-rank approximation: A framework for incomplete multiview learning. *IEEE Transactions on Cybernetics*.

Tierney, S.; Gao, J.; and Guo, Y. 2014. Subspace clustering for sequential data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1019–1026.

Tucker, L. R. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3): 279–311.

Vidal, R. 2011. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2): 52–68.

Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416.

Wang, Q.; Chen, M.; Nie, F.; and Li, X. 2018a. Detecting coherent groups in crowd scenes by multiview clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1): 46–58.

Wang, Y.; Wu, L.; Lin, X.; and Gao, J. 2018b. Multiview spectral clustering via structured low-rank matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10): 4833–4843.

Wu, F.; Hu, Y.; Gao, J.; Sun, Y.; and Yin, B. 2015. Ordered subspace clustering with block-diagonal priors. *IEEE Transactions on Cybernetics*, 46(12): 3209–3219.

Xie, Y.; Tao, D.; Zhang, W.; Liu, Y.; Zhang, L.; and Qu, Y. 2018. On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *International Journal of Computer Vision*, 126(11): 1157–1179.

Xu, C.; Tao, D.; and Xu, C. 2015. Multi-view learning with incomplete views. *IEEE Transactions on Image Processing*, 24(12): 5812–5825.

Yang, C.; Robinson, D.; and Vidal, R. 2015. Sparse subspace clustering with missing entries. In *International Conference on Machine Learning*, 2463–2472.

Yu, J.; Li, C.; Zhao, Q.; and Zhao, G. 2019. Tensor-ring nuclear norm minimization and application for visual data completion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3142–3146.

Yu, J.; Zhou, G.; Li, C.; Zhao, Q.; and Xie, S. 2020. Low tensor-ring rank completion by parallel matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*.

Yu, J.; Zhou, G.; Zhao, Q.; and Xie, K. 2018. An effective tensor completion method based on multi-linear tensor ring decomposition. In *IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1344–1349.

Zhang, C.; Hu, Q.; Fu, H.; Zhu, P.; and Cao, X. 2017. Latent multi-view subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4279–4287.

Zhang, H.; Zhai, H.; Zhang, L.; and Li, P. 2016. Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6): 3672–3684.

Zhang, Y.; Jiang, Z.; and Davis, L. S. 2013. Learning structured low-rank representations for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 676–683.

Zhao, H.; Liu, H.; and Fu, Y. 2016. Incomplete multi-modal visual data grouping. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2392–2398.

Zhao, Q.; Zhou, G.; Xie, S.; Zhang, L.; and Cichocki, A. 2016. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*.

Zheng, R.; Li, M.; Liang, Z.; Wu, F.-X.; Pan, Y.; and Wang, J. 2019. SinNLRR: a robust subspace clustering method for cell type detection by non-negative and low-rank representation. *Bioinformatics*, 35(19): 3642–3650.

Zhou, P.; Lu, C.; Feng, J.; Lin, Z.; and Yan, S. 2019. Tensor low-rank representation for data recovery and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5): 1718–1732.